STRUCTURES FOR LUT-BASED ARITHMETIC IN PLDS

BACKGROUND OF THE INVENTION

[0001] The present invention is in the field of programmable logic devices (PLD's) and, more particularly, is related to the use of look-up tables (LUTs) in logic elements (LEs) of PLD's where not only are multiple stages of arithmetic performed in each LUT but, also, the LUTs are configurable to accomplish multiple non-arithmetic functions with reduced input sharing requirements.

[0002] A common use for LE logic in a PLD is to perform arithmetic operations. LEs of conventional PLDs provide arithmetic functionality with 4-input LUTs. (We refer to a LUT with "x" inputs as an "x-LUT.") Figure 1 is a block diagram of a conventional 4-LUT 100.

[0003] The 4-LUT 100 can be thought of, functionally, as two 3-LUTs 102 and 104. The 4-LUT 100 performs arithmetic, with the ability to add two bits in each 3-LUT 102 and 104. A multiplexer 106 selects between the outputs of the 3-LUTs 102 and 104, using the fourth data input D to the 4-LUT 100 as a select line to control the output of the multiplexer 106. Conventional implementations of LUT arithmetic use one 3-LUT of a 4-LUT to generate the sum from two summand signals and the carry_in of the previous stage. The other 3-LUT generates the carry_out of the stage from the same three input signals (i.e. the two summand signals and the carry_in of the previous stage). A benefit of such a scheme is that an arithmetic sum output is generated by the same output circuitry that may be used to generate separate arithmetic functions. This allows arithmetic to be performed without requiring any additional multiplexers for selection between different outputs. Figure 2 is a block diagram of a LUT 200, illustrating the LUT 100 configured according to the conventional approach.

BRIEF SUMMARY OF THE INVENTION

[0004] A programmable logic device (PLD) includes a plurality of logic array blocks (LAB's) connected by a PLD routing architecture. At least one LAB includes a logic element (LE) configurable to arithmetically combine a plurality of binary input signals in a plurality of stages.

[0005] The LE comprises look-up table (LUT) logic having K inputs (a "K-LUT"). The K-LUT is configured to input the binary input signals at respective inputs of the K-LUT logic cell and to provide, at a plurality of outputs of the K-LUT logic cell, respective binary result signals indicative of at least two of the plurality of stages of the arithmetic combination of binary input signals.

[0006] An input line network includes a network of input lines, the input lines configurable to receive input signals from the PLD routing architecture that represent the binary input signals and to provide the input signals to the K-LUT. An output line network includes a network of output lines, the output lines configured to receive, from the K-LUT, output signals that represent the binary result signals and to provide the output signals to the PLD routing architecture.

[0007] The described LUT's can perform arithmetic efficiently, as well as non-arithmetic functions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 is a block diagram of a conventional 4-LUT 100;

[0009] FIG. 2 is a block diagram of a LUT 200, illustrating the LUT 100 configured according to a conventional approach;

[0010] FIG. 3 illustrates a fracturable 6-LUT 300;

[0011] FIG 4 is a block diagram of a LUT 400, illustrating the LUT 300 configured according to a particular approach;

[0012] FIG 5 is a block diagram of a LUT 500, illustrating how the LUT 300 (Figure 3) can be enhanced, including the input multiplexers 502 and 504.

[0013] FIG 6 is a block diagram of a LUT 600, illustrating how the LUT 500 may be configured according to a particular approach;

[0014] FIG 7 shows such a particular example 700 of the LUT 300 with additional input multiplexers 702, 704, 706 and 708 provided;

[0015] FIG 8 illustrates a LUT 800 that is similar to the LUT 700, configured to accomplish arithmetic in a manner similar to that of the LUT 600 (Figure 6) but without any input sharing requirement;

[0016] FIG 9 illustrates a LUT 900 which shows the 4-LUTs 302, 304, 306 and 308 as the component 3-LUTs;

[0017] FIG 10 illustrates a LUT 1000 in which this issue is addressed by exchanging where certain signals are used in the top and bottom half of the LUT, as well as by moving the fractured inputs (C_1, C_2, D_1, and D_2) to earlier LUT stages;

[0018] FIG 11 illustrates a LUT 1100 that includes two additional outputs Arith0 and Arithl (which are used when doing LUT-based arithmetic), and two multiplexers 1102 and 1104, that feed those respective outputs with the sum;

[0019] FIG 12 illustrates a LUT 1200 configured in a manner that "undoes" an input swap to remove a speed penalty resulting therefrom.

DETAILED DESCRIPTION OF THE INVENTION

[0020] In accordance with a broad aspect of the invention, LEs are provided with k-LUT's (typically, but not limited to, where $k \geq 4$), wherein multiple stages of arithmetic may be performed in each k-LUT. The k-LUT may be, but is not required to be, fracturable. A fracturable LUT is a LUT that includes "extra" inputs, such that input sharing requirements are reduced (while not necessarily precluding input sharing, however).

[0021] The principle of fracturable LUTs is illustrated in Figure 3, where a fracturable 6-LUT 300 is shown. What makes the 6-LUT 300 fracturable is a 2:1 output multiplexer 310 (circled in the figure). When used to perform an arithmetic function, the output signal z1 is a function of all 6 inputs (A through F). When used to accomplish separate logic functions, the output multiplexer 310 is configured such that the z0 output, which is provided from the portion of the LUT 300 including 4-LUTs 302 and 304 provides a function of A, B, C, D, and E while the z2 output, provided from the 2:1 multiplexer 310, provides a function of A, B, C, D, and F. In this way, the LUT 300 can implement two different 5-input logic functions as long as those two logic functions share four inputs (i.e. A, B, C, and D). The input multiplexers

312, 314 select between carry signals and the D input to allow for arithmetic operation (by allowing the second and fourth 4-LUTs 304 and 308 to steal the D signal as a carry_out signal).

[0022] When using fracturable LUTs, there are multiple outputs from each logic element, allowing the fracturable LUT to perform more than one stage of LUT-based arithmetic without requiring any additional output logic. For example, a logic element with a fracturable K-LUT can use one (K-1)-LUT to accomplish one stage of arithmetic and the other (K-1)-LUT to accomplish a second stage of the arithmetic. Each of these (K-1)-LUTs can be thought of as having two (K-2)-LUTs. One of these (K-2)-LUTs can be used to provide the sum for two functions of no more than K-3 distinct input signals in total (the sum being a function of those K-3 inputs and the carry_in from the previous stage). The other (K-2)-LUT can be used to provide the carry_out of the stage (as a function of the same K-3 inputs and the carry_in of the previous stage).

[0023] Figure 4 is a block diagram of a LUT 400, illustrating the LUT 300 configured according to the approach just discussed with reference to Figure 3. As illustrated in Figure 4, the output multiplexer 316 (Figure 3) is configured to provide the output of 4-LUT 302, and the output multiplexer 318 (Figure 3) is configured to provide the output of 4-LUT 306. In some embodiments, the LUTs are fractured even further to have N outputs, to perform N stages of arithmetic in each such fracturable LUT.

[0024] As discussed above, the output multiplexer 310 (Figure 3) is provided to reduce the input sharing requirements when the LUT 300 is used to accomplish two functions. In that sense, the output multiplexer 310 is not material when the LUT 300 is used to accomplish arithmetic, as configured in Figure 4. Put another way, the output multiplexer 310 contributes to the flexibility of the LUT 300 for use to implement functions (lessening the input sharing requirements for the functions), such that the LUT 300 is also configurable as shown in Figure 4 to accomplish arithmetic.

[0025] The scheme just described with reference to Figure 3 and Figure 4 allows each stage of arithmetic to implement the sum for two functions of no more than 3 distinct inputs in total. All 3 distinct inputs are shared between the two stages that a single LUT can implement -- A, B, and C in Figure 4. That is, the LUT uses shared inputs to implement multiple logic functions, and only those pairs of (K-1)-input functions that meet this sharing requirement can be

put in the same LUT. This property of fracturable LUTs makes it desirable to minimize the amount of sharing that is required between two stages of arithmetic, which allows for a wider range of arithmetic functions.

[0026] As discussed below, this minimization of input sharing can be accomplished with input multiplexers that allow portions of the LUT to use otherwise unused inputs, to reduce the sharing requirement. Figure 5 is a block diagram of a LUT 500, illustrating how the LUT 300 (Figure 3) can be enhanced, including the input multiplexers 502 and 504. Figure 6 is a block diagram of a LUT 600, illustrating how the LUT 500 may be configured according to the approach just discussed with reference to Figure 5. In particular, Figure 6 shows the LUT 500 configured to implement sums of no more than three distinct inputs in total, where only two of the three signals (A and B in Figure 6) are shared between stages of arithmetic in the same LUT. This is in contrast to the three input signals that are shared in the Figure 4 LUT 400.

[0027] Additional input multiplexers may be provided to allow for more than K inputs to a fracturable K-LUT. For example, Figure 7 shows such a particular example 700 of the LUT 300 with additional input multiplexers 702, 704, 706 and 708 provided to achieve a fracturable 6-LUT with 8 inputs. Figure 8 illustrates a LUT 800 that is similar to the LUT 700, configured to accomplish arithmetic in a manner similar to that of the LUT 600 (Figure 6) but without any input sharing requirement. That is, in Figure 8, the LUTs 302 and 304 perform one stage of arithmetic -- the sum of two functions of no more than three inputs (A, C_1, and D_1). The LUTs 306 and 308 perform another stage of arithmetic -- the sum of two functions of a completely different set of three inputs (C_2, B, and D_2).

[0028] We now discuss embodiments in which the arithmetic performance of a fracturable LUT is increased by providing a special path for the C_{in} signal. That is, with the Figure 7 LUT 700, the fractured inputs (Cl, C2, D1, and D2) are in the middle of the LUT input stages. (By convention, the inputs are labeled such that A is the earliest and, hence, slowest stage. Meanwhile, the input F is the latest and, hence, fastest stage. Because the C_{in} input uses the same input multiplexer as the C_1 and C_2 inputs, the path from C_{in} to C_{out} goes through three levels of multiplexing (the multiplexer to select C_{in}, the C-input stage of the LUT, and the D-stage of the LUT). Figure 9 illustrates a LUT 900 which shows the 4-LUTs 302, 304, 306 and 308 as the component 3-LUTs (902 and 904; 906 and 908; 910 and 912; and 914 and 916).

[0029] In addition, two multiplexers 920 and 922 are provided to provide a more direct input path for the C_{in} signal. With the multiplexers 920 and 922, the path from carry_in to carry_out takes one stage of multiplexing. This scheme, however, still uses additional multiplexers before the LUTs, which can slow down the signals being multiplexed as well as waste area in the LUT. Figure 10 illustrates a LUT 1000 in which this issue is addressed by exchanging where certain signals are used in the top and bottom half of the LUT, as well as by moving the fractured inputs (C_1, C_2, D_1, and D_2) to earlier LUT stages.

[0030] A disadvantage of using the structure illustrated by Figure 10 as opposed to the structure illustrated by Figure 9 is that a full 6-input LUT has both inputs A and B having the delay of the slowest LUT stage. This is because input A feeds the slowest stage in the top half, while input B feeds the slowest stage in the bottom half. Furthermore, the fractured inputs are in slower LUT stages. The fractured inputs being in slower LUT stages can be an advantage as well, a tradeoff that can be considered when deciding on which structure to use.

[0031] Although the carry_in to carry_out path has been shortened in Figure 9 and Figure 10, the carry_in to sum path is still relatively long. That is, after carry_in is applied (to the C input stage in Figure 9 and to the D input stage in Figure 10), it travels through at least 3 multiplexer stages before reaching the LUT output as the sum signal. By creating an extra output for the LUT, a technique similar to that used to shorten the carry_in to carry_out path can be used to speed up the carry_in to sum path. Figure 11 illustrates a LUT 1100 that includes two additional outputs Arith0 and Arith1 (which are used when doing LUT-based arithmetic), and two multiplexers 1102 and 1104, that feed those respective outputs with the sum.

[0032] Adding extra outputs to the LUT is particularly effective for a fracturable LUT because there are already multiple outputs from the LUT, and output multiplexers are used to select from the multiple outputs. This technique is generally applicable, however, and can be used for regular non-fracturable LUTs, particularly if the additional output multiplexing cost is acceptable to achieve a shorter carry path.

[0033] A further disadvantage of the Figure 10 LUT 1000 is that the inputs A and B both have the largest LUT delay when a 6-input LUT is used because the output of the 6-input LUT is a function of both A and B.. That is, because A and B are in opposite orders in the LUT inputs, both A and B are used in the slowest stage of either the top-half LUT or bottom-half

LUT. Thus, the worst-case delay is applied to both when calculating the delay of the circuit. The reason this is done is to allow the top half and bottom half to have different arithmetic functions even though the A and B inputs are not fractured.

[0034] Figure 12 illustrates a LUT 1200 configured in a manner that "undoes" this swap to remove the speed penalty. However, an associated cost is that the two arithmetic functions now share one input between them. This can be an acceptable cost for a LUT that is large enough that requiring one shared input does not significantly hurt the flexibility of functions that can be implemented.

[0035] Although particular embodiments have been described in detail, various modifications to the embodiments described herein may be made without departing from the spirit and scope of the present invention, thus, the invention is limited only by the appended claims. For example, the disclosed embodiments may be employed in conjunction with LUTs that use LUT mask sharing as disclosed in US Patent Application No. 10/351,026, filed January 24, 2003 and assigned to the assignee of the present patent application.